

C3. Facelt Program Commands

Facelt (and FaceSt) supports commands dealing with program initialization and management of events.

In addition to the commands presented below, standard label ID numbers can also be used as direct Facelt commands if Facelt (not FaceSt) is in use and the front window is not modal. The following line, for example, would execute the standard "Copy" menu item:

```
Facelt(nil,14,0,0,0); #14 = Copy
```

Name Number Parameters & Variables used

DoInit -61 a,b,c,d,uName,uResult

This command was first described in the "Initializations" topic in the "Startup" menu. The additional information presented here supplements the original description and only applies if using Facelt or FaceSt.

If using Facelt or FaceSt ($c \geq 0$), then parameter a can be used to set various Facelt or FaceSt options that affect program operation, and parameter c can be used to disable certain default operations performed by Facelt or FaceSt when DoInit is called.

a = sum of following constants (bit flags stored in fFlags):

1* = change cursor to deck of cards when returning control from DoLoop (can be helpful during development)

2 = return control to program whenever active window is changed (returns with uMenuID = 1100, uMenuItem = 2, uResult = 0)†

4 = return control to program whenever clipboard is changed (returns with uMenuID = 1100, uMenuItem = 4, uResult = 0)

8* = validate all non-ViewIt windows that require updating instead of hiding these windows (useful in some debugging environments that open status windows)

16* = disable all auto-saving on Quit or Transfer (no Save Document notices are posted to FCMD modules)

32* = hide all ViewIt windows when switching to background under MultiFinder or System ≥ 7 (overrides "Hide on Switch" in Window dialog)††

64 = don't show colors in ViewIt windows (overrides "Bits/Pixel Minimum" in Window dialog)

128 = disable all switching under System ≥ 7 when a modal window is open or a window is in edit mode (overrides "Allow Modal Switch" option in Window dialog)

512* = return all Apple events that are not required events handled by Facelt (returns uMenuID = 0 and the Apple event in fEvent), otherwise Facelt processes such events along with the required events by calling "AEProcessAppleEvent"

1024* = return control to program before ExitToShell or transfer is executed by Facelt (see discussion below)††

b = kilobytes of extra stack space (see discussion of this parameter in "Initializations" in "Startup" topics)

c = sum of following constants (bit flags):

1 = skip toolbox initializations (done in FaceProcXY)

2* = skip auto-initialization & insertion of main menus

4* = skip "About ProgramName" standard item setup

32* = do not auto-open any splash screen

64* = do not auto-open ViewIt Help window (this window) when ViewIt editing resources are available

where the constants that are followed by "*" correspond to functionality not supported by FaceSt, and "return control" in the context of FaceSt means that a pseudo-menu event is posted.

When returning from DoInit, some programs need to know whether the user is attempting to open or print one or more program data files "from the Finder" so that, if not, then they can open a default window. If using Facelt, this information is returned in uResult: 0 = no files, 1 = one or more files are being opened from the Finder. If files are being opened from the Finder, then one message for each such file will be returned later from DoLoop (see description below).

† Closing a modal window can also result in this message being posted, so do not assume that this message is only posted when the active window changes. If necessary, track the current program context in a program variable and only respond to this message if the new context truly differs from that saved by your program.

†† This bit option can be reset by the program at any time (using a call such as "BitSet(@fFlags,26)" or "BitClr(@fFlags,26)"), and could be used to provide a user with the option of hiding all windows when switching out.

††† Adding 1024 to a causes Facelt to return control to the program just before executing ExitToShell (if Quit chosen) or transferring to another program (if Transfer chosen). The message returned has uMenuID = 1100, uMenuItem = 1024, uResult = 1 (Quit) or 2 (Transfer), and uName = name of program to transfer to. Any subsequent call to Facelt will then cause the program to immediately quit or transfer according to the contents of uName: if uName is empty, then the program quits; if uName is not empty, then the program transfers to the named application. Warning: If you reset uName to transfer to a different application, be certain that the current directory contains that application.

Also note that adding 1024 does not replace the use of negative label numbers (" #-106") in menu items, since these can be used to get control before Facelt goes through its Quit or Transfer- related operations (opening of the Transfer dialog, prompts to save changes, etc.), whereas bit value 1024 returns control after these operations are performed and the Quit or Transfer cannot be cancelled.

DoLoop 0 a

Used to enter Facelt's main event loop which automatically handles most events. Control remains with Facelt's event loop until an event occurs that "belongs" to your program. Info about such events is then returned in uMenuID, uMenuItem, uResult, uString, and other fRec variables. One such event, for example, is the open/print a file "from the Finder" message described in the "Finder Resources" topic:

uMenuID = 1100

uMenuItem = 512

uResult = 1 (open) or 2 (print)

uString = file type

uName = file name

& current directory set that containing the file

which is returned from DoLoop whenever the user attempts to open or print a file from the Finder.

OPTIONS: Parameter a can optionally be used to have control returned to the program immediately after calling DoLoop. If a = -1, then control is returned after updating all windows, the active window status, etc. (same as calling DoUpdt w/ d = -1). If a = -2, then control is only returned after processing all pending events. In this case, if no event is the cause of control being returned, then uMenuID and fEvent.what will be zero. These options are similar to the options supported by parameter b when using MdIWnd to handle events in modal ViewIt windows. (The "Background Processing" topic contains a further discussion of these options.)

PORT NOTE: DoLoop always resets the current port to the active window unless the last event returned was a raw Apple event (the exception for Apple events was necessary to support interapplication communication via the ScriptIt module).

SLEEP NOTE: While executing DoLoop, Facelt "sleeps" for a period of time determined by the variable fFrontSleep when in the foreground and fBackSleep when in the background under MultiFinder or System 7. This sleep time will affect any operation that relies on "idle" time to get things done. This includes the use of parameter a = -2 to get control back from DoLoop, and ViewIt controls such as QuickControl which relies on the "hook" message when playing movies. To give more time to such operations (at price of slowing background

operations occurring in other programs), the default sleep times of 6 (front) and 8 (back) ticks can be made smaller. To do this you need to reset both the current sleep time, fSleep, and one or both of the sleep parameters fFrontSleep and fBackSleep that are used to reset fSleep when switching occurs. The QuickControl demo, for example, contains the following lines after DoInIt:

```
fRec.fSleep := 0;  
fRec.fFrontSleep := 0;
```

which results in better performance when playing movies. Be careful that the resulting fFrontSleep \neq fBackSleep since the foreground or background state of the program is tracked by comparing fSleep with these variables.

NOTE: DoLoop is not supported by FaceSt, and should not be called when a modal window is open.

DoEvt -51 fEvent

Asks Facelt or FaceSt to handle the event passed in fEvent. If using Facelt, then DoEvt is used to pass events to Facelt that are encountered while programs are doing background processing (discussed in "Background Processing" topic in Facelt guide).

If using FaceSt, then DoEvt is used to pass FaceSt all ViewIt window-related events that are not handled by the main program (described in "Hybrid Programs" topic).

NOTE: DoEvt should not be called when a modal window is open.

DoMenu -52 a,b,uMenuID,uMenuItem

Forces Facelt or FaceSt to handle the menu event passed in parameters a (menuID) and b (menu item number). If a and b are zero, then Facelt and FaceSt look in uMenuID and uMenuItem for the identity of the menu item to execute. DoMenu is typically used with FaceSt to pass it selections from main program menus. If the menu selection does not correspond to a standard menu item, then FaceSt reposts the menu event as a message returned by GetMsg.

NOTE: DoMenu should not be called when a modal window is open.

GetMsg -55 [none]

FaceWare modules return special messages and events to the main program by placing them in a private queue with the utility command PstEvt (see "Other Utilities" topic for more info). When using Facelt, messages from the private queue are automatically processed and returned from DoLoop as menu or pseudo-menu events. When using FaceSt, the main program must call GetMsg to get the next message in the queue if the queue is not empty (see "Hybrid Programs" topic for details).

NOTE: GetMsg is not supported by Facelt.